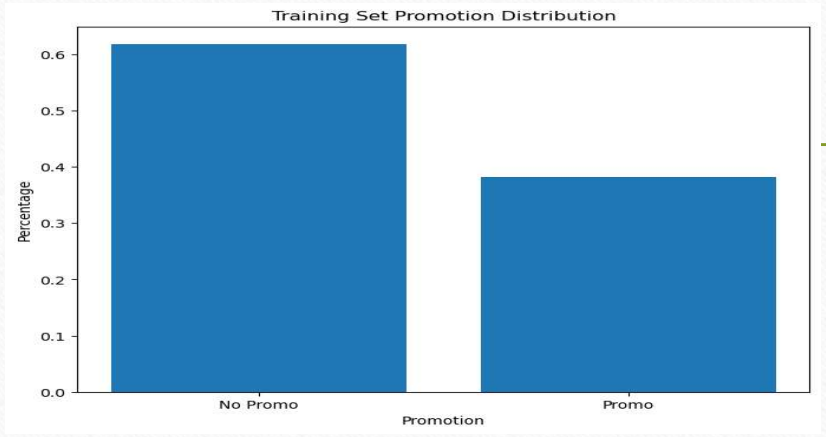
Rossmann Pharmaceuticals
Exploration of customer purchasing behavior

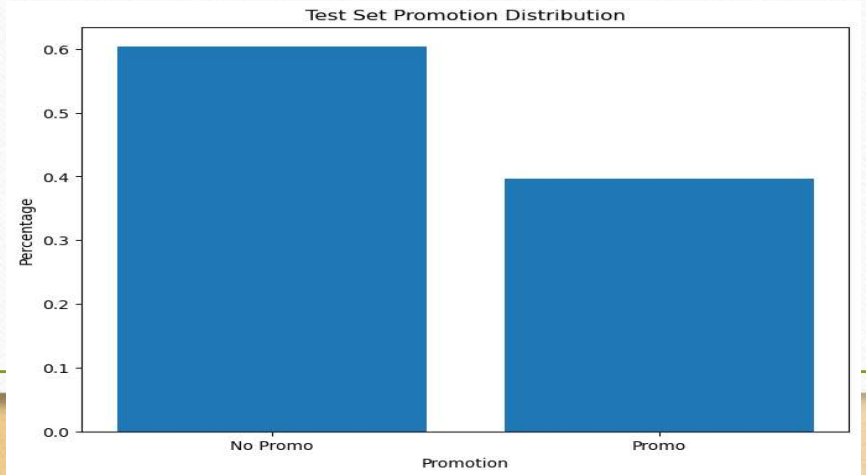
Yodahe Teshome

Email-Jodahe2001@gmail.com

Git Link For The Project-> https://github.com/jodahe1/Rossmann-Pharmaceuticals-.git

Check for distribution in both training and test sets - are the promotions distributed similarly between these two groups?





As we can see on the image they are slightly different

In the training set, approximately 38.2% of the records have a promotion (Promo=1), while around 61.8% do not have a promotion (Promo=0).

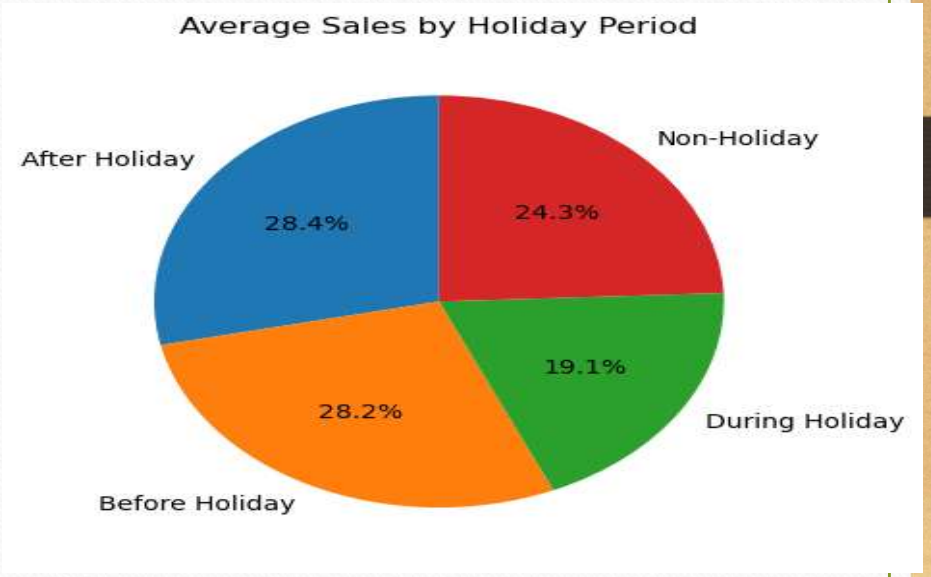
In the test set, approximately 39.6% of the records have a promotion (Promo=1), while around 60.4% do not have a promotion (Promo=0).

Although the distributions are not exactly the same, they are relatively similar, with both sets having a higher proportion of records without promotions (Promo=0).

Check & compare sales behavior before, during, and after holidays

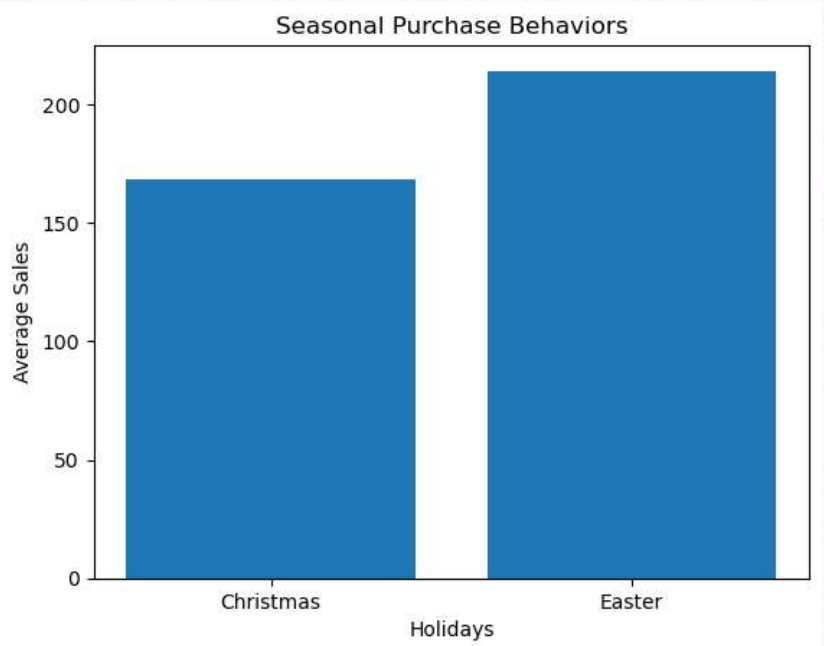
- As we can see on the chart sales is high Before holiday many things contribute for this like peoples want to go to holiday by being prepared or there might be some Discounts and more but we can conclude it's high during this period .

So we can learn from this that we need to increase stock at that time.



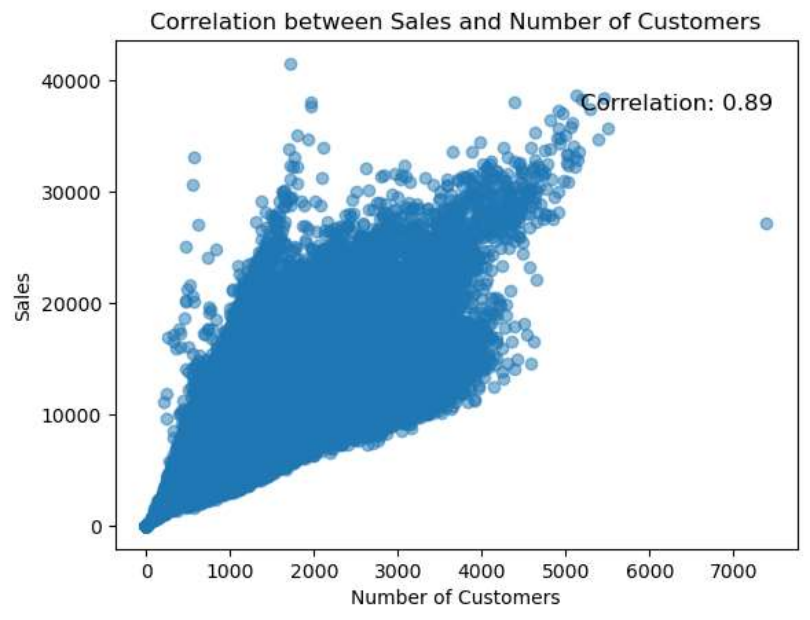
Find out any seasonal (Christmas, Easter, etc) purchase behaviors,

- We can see there is seasonal Behavior when is comes to purchase since the Average sales is not same in Christmas And easter peoples tends to buy more During easter.

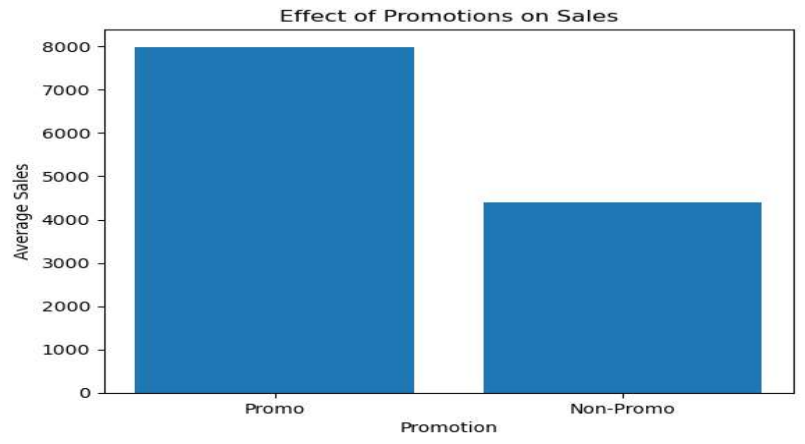


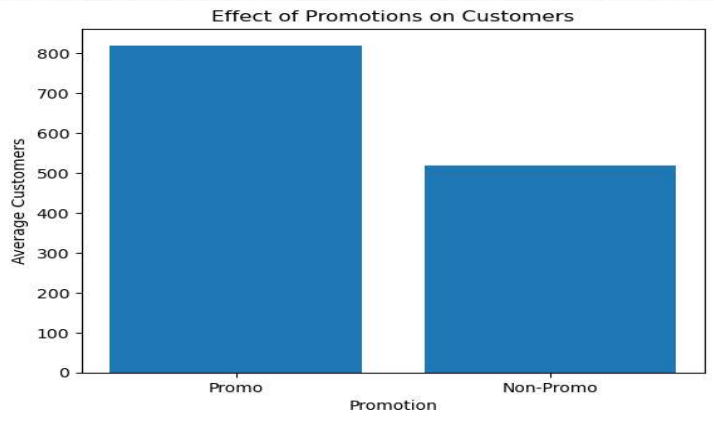
What can you say about the correlation between sales and the number of customers?

- Based on the correlation value of 0.895, we strong positive correlation between sales and the number of customers. This means that as the number of customers increases, the sales also tend to increase. The correlation value of 0.895 suggests a strong linear relationship between these variables.



How does promo affect sales? Are the promos attracting more customers? How does it affect already existing customers?





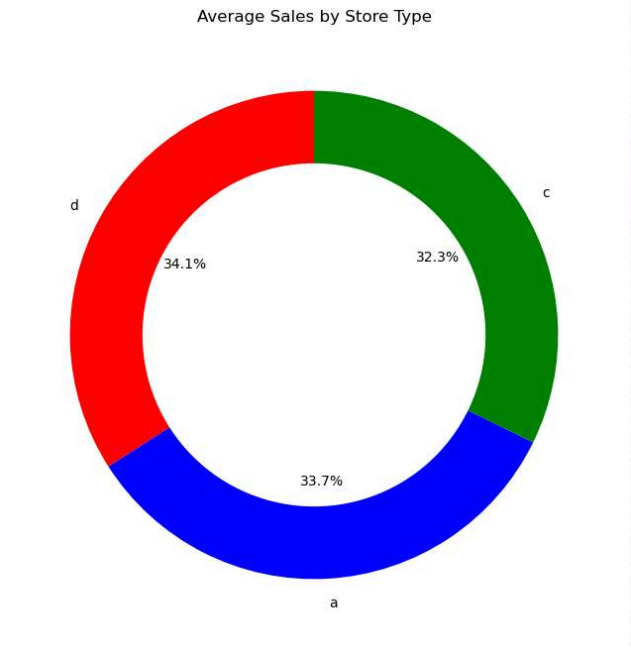
The average sales during promotional days (avg_sales_promo) are approximately 7991.15, while the average sales on non-promotional days (avg_sales_non_promo) are around 517.82. This suggests that promotions have a positive impact on sales, as the average sales during promos are significantly higher than on non-promo days.

In terms of attracting more customers, the average number of customers during promotional days (avg_customers_promo) is approximately 820.10, while the average number of customers on non-promotional days (avg_customers_non_promo) is approximately 517.82. This indicates that promotions do attract more customers, as the average number of customers during promos is higher than on non-promo days.

Therefore, based on the provided data, we can conclude that promotions positively impact sales by attracting more customers.

Could the promos be deployed in more effective ways? Which stores should promos be deployed in?

- Based on the average sales by store type, it appears that Store Types "d", "a", and "c" have higher average sales compared to other store types.
- Therefore, deploying promotions in these store types could be more effective in boosting sales.



Trends of customer behavior during store opening and closing times

- We can clearly see in the chart that there i

Relatively 0 activity during closing time but

There is more activity on opening times

Compared to closing time and it also varies

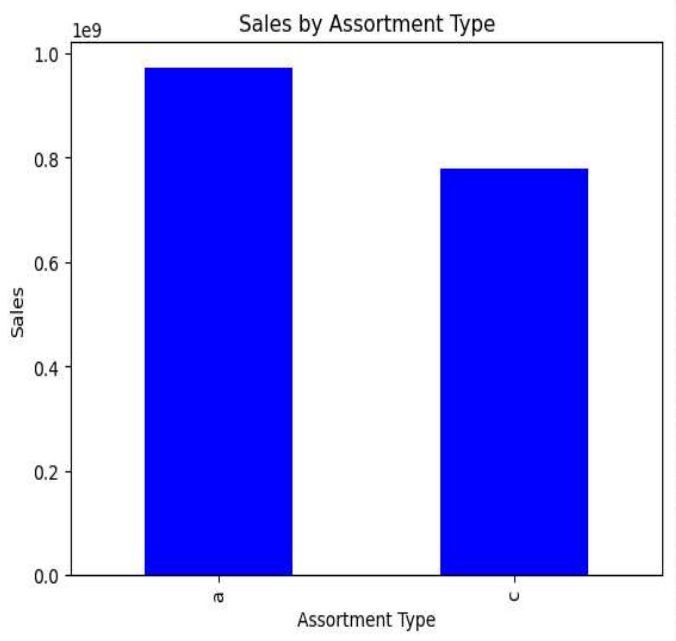
From day to day it's high on Sunday and

Relatively low Saturday.

Check how the assortment type affects sales

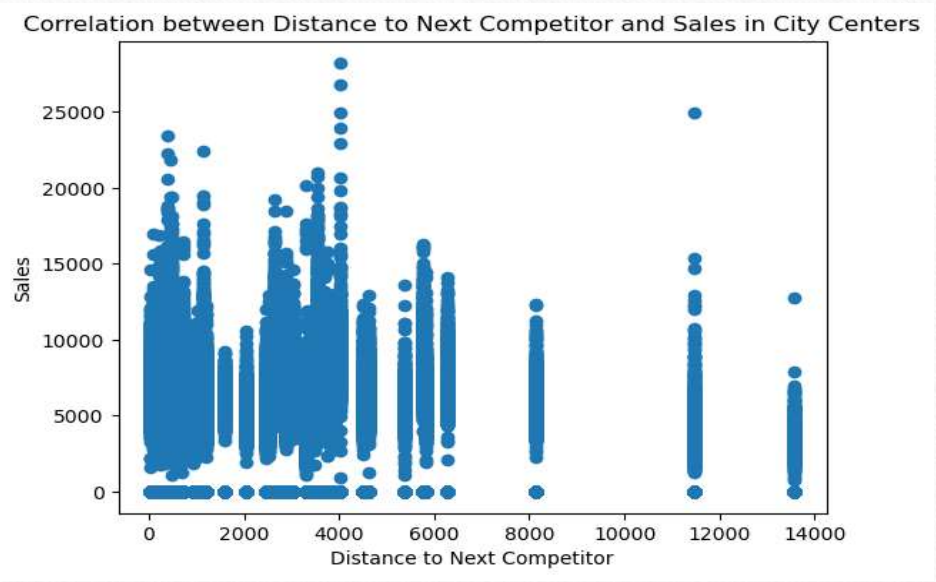
- we can see that the assortment type 'a' has a total sales value of 971,834,451, while the assortment type 'c' has a total sales value of 779,193,689. From this information, we can conclude that assortment type 'a' has higher sales compared to assortment type 'c'.

This indicates that the assortment type does affect sales, with assortment type 'a' having a higher impact on sales compared to assortment type 'c'.



How does the distance to the next competitor affect sales? What if the store and its competitors all happen to be in city centers, does the distance matter in that case?

- It did some correlation analysis and Based on the correlation value of -0.084, there seems to be a weak negative correlation between the distance to the next competitor and sales in city centers. This suggests that as the distance to the next competitor increases, sales may slightly decrease, although the relationship is not very strong. However, it's important to note that correlation does not imply causation. Other factors may also influence sales in city centers, and the distance to the next competitor alone may not be the sole determinant.



Task-2

- Starting from this slide I would like to share what I did using deep learning.
- I have notes and images which shows what I

Try to do.

Pre-Processing

- The first thing I do here is trying to change non-numeric column to numeric one hence am going to use them for ML.
- And I checked Null values , to avoid error on result

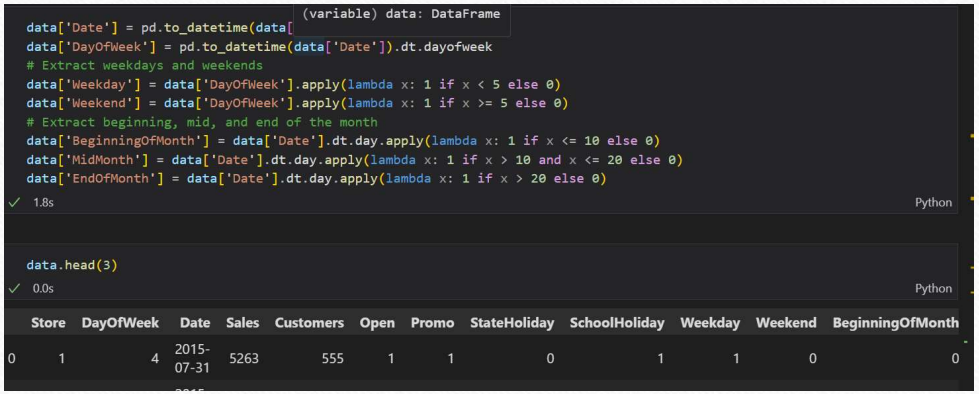
```python
# Define the mapping dictionary
holiday_mapping = {"a": 1, "b": 2, "c": 3, "0": 0}

# Map the values in the "StateHoliday" column to integers
data["StateHoliday"] = data["StateHoliday"].map(holiday_mapping)
```

✓ 0.0s Python

```python
# Check for NaN values
print(data.isna().sum())
```

Extracting Features And Scaling

- Here as I tried to show on the image I tried to extract some basic features from date column and I use `StandardScaler()` to scale all numeric columns.



```
data['Date'] = pd.to_datetime(data[
(variable) data: DataFrame
data['DayOfWeek'] = pd.to_datetime(data['Date']).dt.dayofweek
# Extract weekdays and weekends
data['Weekday'] = data['DayOfWeek'].apply(lambda x: 1 if x < 5 else 0)
data['Weekend'] = data['DayOfWeek'].apply(lambda x: 1 if x >= 5 else 0)
# Extract beginning, mid, and end of the month
data['BeginningOfMonth'] = data['Date'].dt.day.apply(lambda x: 1 if x <= 10 else 0)
data['MidMonth'] = data['Date'].dt.day.apply(lambda x: 1 if x > 10 and x <= 20 else 0)
data['EndOfMonth'] = data['Date'].dt.day.apply(lambda x: 1 if x > 20 else 0)
✓ 1.8s                                                                    Python
```

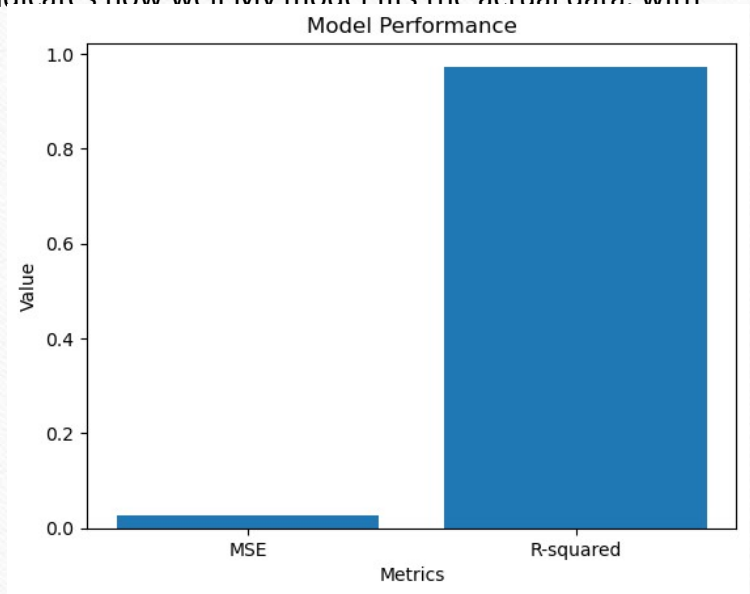
```
data.head(3)
✓ 0.0s                                                                    Python
```

	Store	DayOfWeek	Date	Sales	Customers	Open	Promo	StateHoliday	SchoolHoliday	Weekday	Weekend	BeginningOfMonth
0	1	4	2015-07-31	5263	555	1	1	0	1	1	0	0

Building models

- I use sklearn pipelines to build my model since using pipeline gave me modularity and efficiency , an I tried to calculate Mean Squared Error(0.0272838181610939) and R-squared Score(0.972655077324889 3)
- In the next slide I explain the meaning of each term means .

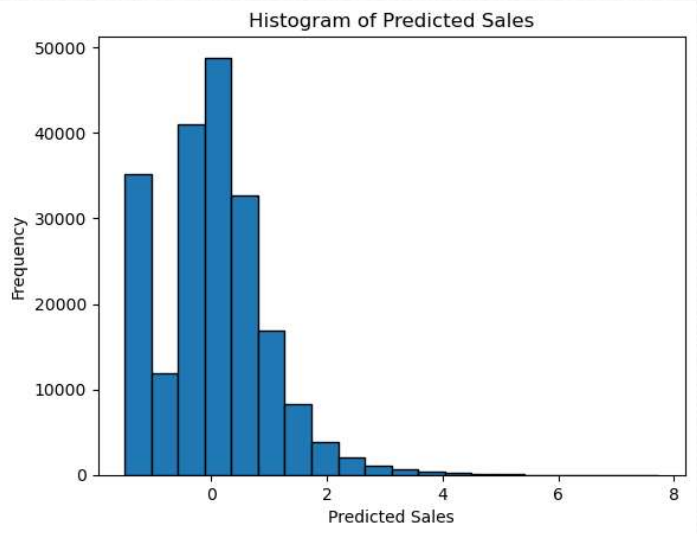
- The Mean Squared Error (MSE) value of 0.02728381816109939 indicates how well My model fits the actual data, with lower values indicating better performance. In this case, the low MSE suggests that your model's predictions are close to the actual values.
- Additionally, the R-squared score of 0.9726550773248893 is a measure of how well my model explains the variability of the data. It ranges from 0 to 1, with a higher value indicating a better fit. In this case, the high R-squared score suggests that your model explains approximately 97.3% of the variability in the data, which is quite good.



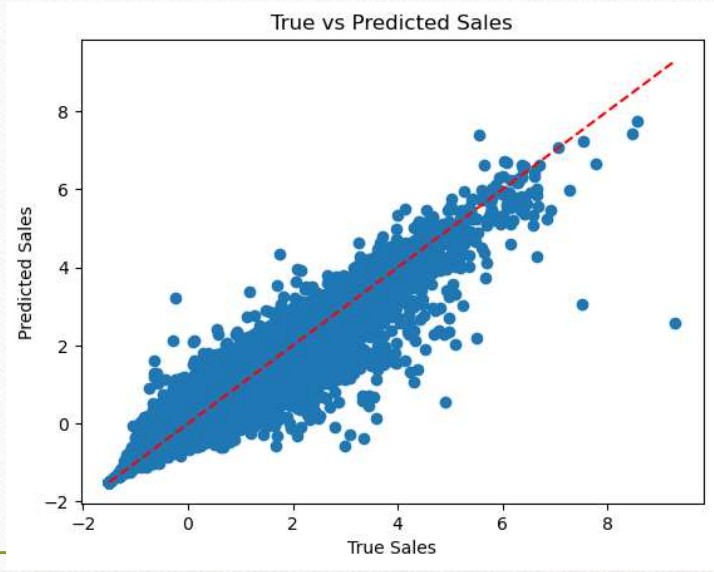
Discounted Sales Loss

- I use a custom loss function called mean_absolute_percentage_error that calculates the MAPE between the true sales values (y_true) and the predicted sales values (y_pred). The MAPE is then printed alongside the Mean Squared Error and R-squared score.
- By using MAPE as the loss function, we can evaluate the model's performance in terms of the percentage error, which provides a more intuitive understanding of the prediction accuracy. It allows us to easily interpret the average percentage deviation of the predicted sales from the true sales values. A lower MAPE indicates a better model fit.
- The value I got from the model is Mean Squared Error: 0.027248994755304307 R-squared Score: 0.972689978720769 Mean Absolute Percentage Error: 103.46544127185126

- So the predicted sales from using the models are shown on the following histogram .



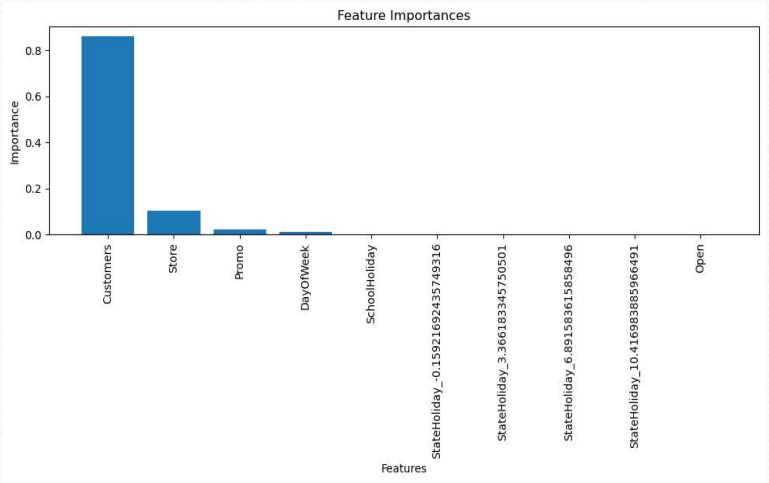
- And this scatter plot shows predicted sales vs the true (actual)sales.



Post Prediction Analysis

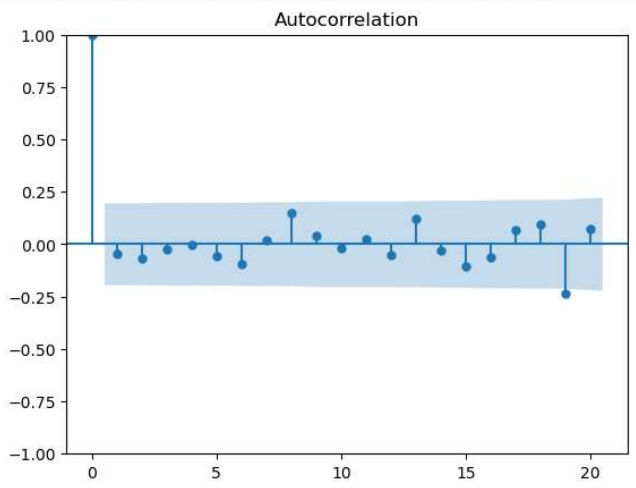
- I try to shows post prediction analysis by evaluating feature importance's .

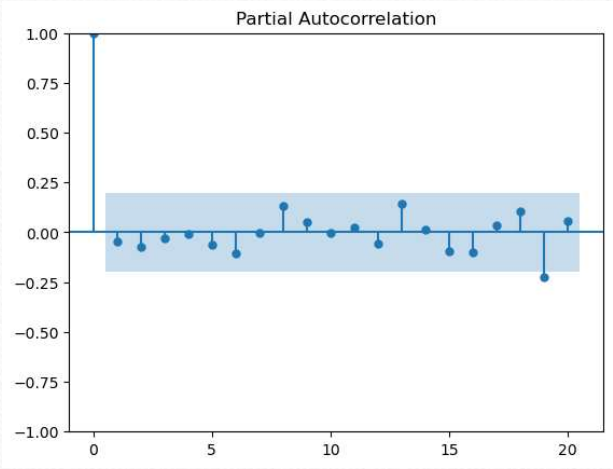
as this image shows us customer is the Most important feature we have so when We make business decisions we need to consider Customers perspective .



LSTM

- I compute Autocorrelation (ACF) and partial autocorrelation (PACF) first to identify the appropriate lag values for My LSTM model.





Transform time series data into supervised learning data

- I use The function which you can see in the image to transform the time series data to supervised learning before I scale it and building mu model .

```python
#  Transform time series data into supervised learning data
def create_supervised_data(data, n_steps):
    X, y = [], []
    for i in range(len(data)-n_steps):
        X.append(data[i:i+n_steps, 0])
        y.append(data[i+n_steps, 0])
    return np.array(X), np.array(y)

n_steps = 10  # Number of previous time steps to consider
X, y = create_supervised_data(differenced_data, n_steps)
```

Python

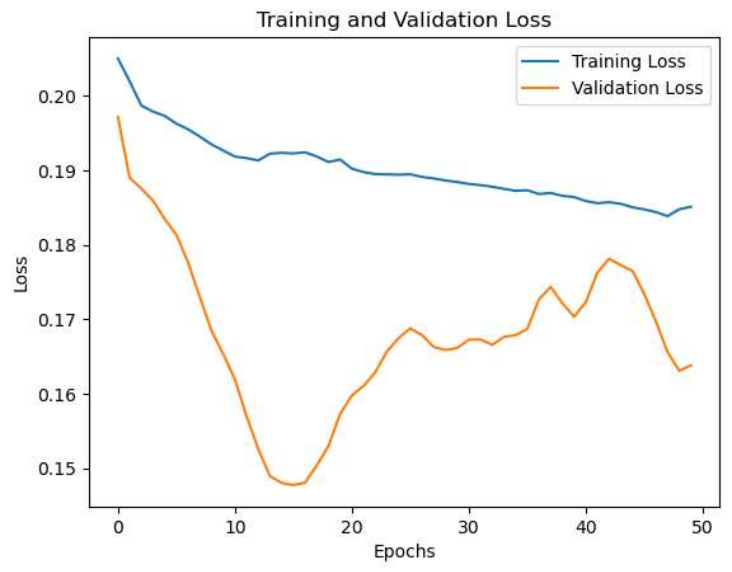
```python
# Step 6: Scale the data
scaler = MinMaxScaler(feature_range=(-1, 1))
X_scaled = scaler.fit_transform(X)
y_scaled = scaler.fit_transform(y.reshape(-1, 1))
```

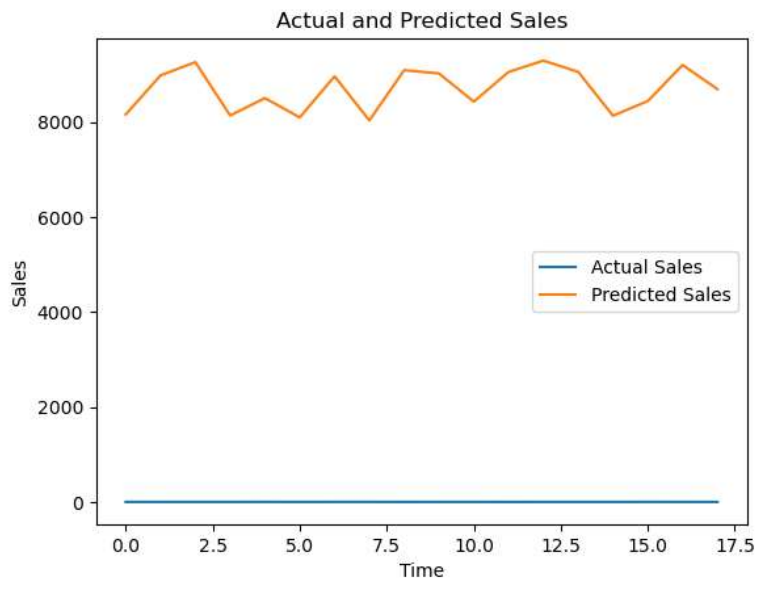
Python

Build the LSTM regression model

- Steps I did to build the model :-
- 1) Split the data into train and test sets
- 2) Train the model
- 3) Make predictions on test data

Model Outputs






THANK YOU